

Multiscale Integrated Simulations of Tokamak Disruptions with Kinetic Runaway Electrons

Dylan Brennan¹, Allen Boozer², Boris Breizman³, Nathaniel Ferraro⁴, Richard Fitzpatrick³, Stefan Gerhardt⁵, Bob Granetz⁶, Eric Hollman⁷, Valerie Izzo⁷, Seung-Hoe Ku⁵, Allen Malony⁸, Manish Parashar⁹, Norbert Podhorszki¹⁰, Mark Shephard¹¹, Edward Startsev⁵

¹*Princeton University*

²*Columbia University*

³*University of Texas, Austin*

⁴*General Atomics*

⁵*Princeton Plasma Physics Laboratory*

⁶*Massachusetts Institute of Technology*

⁷*University of California, San Diego*

⁸*University of Oregon*

⁹*Rutgers University*

¹⁰*Oak Ridge National Laboratory*

¹¹*Rensselaer Polytechnic Institute*

Proposal for Funding from the Scientific Discovery through Advanced Computing (SciDAC):

Multiscale Integrated Modeling for Fusion Energy Science, DE-FOA-0001096

I. OVERVIEW AND MOTIVATION

The overall goal of the proposed research is to develop and implement computational tools to self consistently and accurately simulate the kinetic evolution of runaway electrons coupled to a nonlinear extended magnetohydrodynamic (MHD) simulation of tokamak disruptions. The runaway electrons will be modeled with the XGC1 code [1], and coupled alternately to the MHD codes M3D-C1 [2] and NIMROD [3] through the EFFIS (End-to-end Framework for Fusion Integrated Simulation) framework. This integrated modeling system

will offer the ability to accurately address several critical physics questions associated with runaway electron generation and evolution during tokamak disruptions, and could elucidate mitigation strategies for ITER.

Since the beginning of tokamak experiments, disruptions have both challenged our understanding of the physics of magnetically confined plasmas, and stood directly in the way of progress toward achieving states of higher energy confinement. Though not well understood, disruptions are tolerable on current day experiments. Runaway electron current generation is moderate and thermal deposition to the walls of the experiment, though significant, is relatively benign. The basic physics of runaway current generation has been studied for decades, and progress has been made on the extended MHD simulation of the evolution of disruptions. However, in ITER the runaway electron current generated from a disruption is expected to inductively consume the entire equilibrium current, and the extreme energies deposited on the walls of the experiment are expected to be severely intolerable. The three dimensional electric fields in the MHD instabilities that drive the disruption are expected to generate three dimensional runaway current channels. The evolution of this system, through the thermal collapse of various types of disruption events, is the focus of this proposal.

In advance of ITER operation, we must have a quantitative, predictive understanding of the coupled nonlinear kinetic evolution of runaway current generation and extended MHD evolution of tokamak disruptions, to predict the conditions under which safe and tolerable operation can be achieved.

A. Runaway Electron Theory

The physics issues that will be addressed in this section are: (1) When can runaways exist? (2) When runaways can exist, why in tokamaks the size of ITER, but not in smaller tokamaks, does the current carried by runaways increase exponentially until it becomes the total plasma current? (3) What is characteristic time τ_r for runaways to decay away? (4) Why is the time scale for the transfer of the current from thermal to runaway electrons

given by $\tau_{L/R}$, the decay time for the plasma current due to the resistivity of the thermal electrons? (5) Why is initial seed of runaway electrons generally highly peaked towards the plasma center, which tends to make the runaway current highly peaked. (6) Why is the non-ideal response of a plasma determined by the thermal rather than the runaway electrons even when the current is carried by runaways? (7) Can micro-turbulence reduce the danger of runaway electrons?

1. Electric field requirement

The question of when runaways can exist has a simple answer [?]. The parallel electric field must exceed a critical value, E_r , which is given in Equation (1.1).

A force is required to sustain the energy of a high energy electron against the drag from small-angle collisions with the background electrons. A convenient way to write this force is $eE_{drag}(\gamma)$, where $(\gamma - 1)m_e c^2$ is the electron energy and E_{drag} has units of an electric field. In non-relativistic theory, $\gamma - 1 \ll 1$, the drag force, $eE_{drag}(\gamma)$ scales inversely with the particle velocity squared. In relativistic theory, $\gamma \gg 1$, where the velocity goes to the speed of light, the drag force, $eE_{drag}(\gamma \rightarrow \infty) \equiv eE_r$, is independent of the electron energy [?].

$$E_r = \left(4\pi r_o^2 \frac{m_e c^2}{e} \ln \Lambda \right) n \approx 0.075 n_{20}, \quad (1.1)$$

where E_r has units of Volts/m, n_{20} is the electron density in units of $10^{20}/\text{m}^3$, $\ln \Lambda$ is the Coulomb logarithm, and r_o is the classical radius of an electron, $e^2/(4\pi\epsilon_0 r_o) = m_e c^2$. During normal ITER operations $n_{20} \sim 1$. E_r is not an actual electric field but is a convenient way to write the drag force.

The force exerted by an electric field along the magnetic field, $eE_{||}$, exceeds the drag force when $E_{||} > E_r$. Consequently, when $E_{||} > E_r$ small angle collisions cannot prevent electrons from accelerating to arbitrarily high energy, but at a lower electric field all high energy electrons will eventually slow down to join the thermal distribution.

The precise condition for runaways deviates somewhat from $E_{||} = E_r$, due to velocity space effects, such as pitch angle scattering. Indeed, pitch angle scattering becomes a dom-

inant effect when $(E_{||} - E_r)/E_r \lesssim 1$. Nevertheless, the condition $E_{||} = E_r$ is sufficient for understanding basic issues. When $E_{||} \gg E_r$, subrelativistic electrons can become runaways [?], those with a speed v that satisfies

$$\frac{v^2}{c^2} \geq \frac{E_r}{E_{||}}. \quad (1.2)$$

II. RUNAWAY AVALANCHE

When the electric field along the magnetic field lines exceeds E_r , the drag force of small angle scattering can no longer prevent electrons from reaching arbitrarily high energies. Nevertheless, the typical energy of a runaway electron and their number are determined by what are called knock-on collisions. A single knock-on collision converts a low energy into a runaway electron. Sokolov [?] discussed this effect in 1979, and its importance to ITER-scale tokamaks was pointed out by Hans Fleischmann and co-workers [?]. Nevertheless, the effect is usually called the Rosenbluth avalanche due to the detailed theory worked out by Rosenbluth and Putvinski [?].

The differential cross section [?,?] for an electron of energy $(\gamma - 1)m_e c^2$ to elevate a much lower energy background (or secondary) electron to an energy within the range dW_s of an energy W_s is

$$d\sigma = \frac{\gamma^2}{\gamma^2 - 1} \frac{2\pi r_o^2 m_e c^2}{W_s^2} dW_s, \quad (2.1)$$

where r_o is the classical radius of an electron. The relation between a cross section σ and the rate of collisions with background particles of number density n is $n\sigma v$. When the fast electron is strongly relativistic, so $v = c$, the number of electrons created with an energy above the runaway energy W_r is

$$\begin{aligned} \frac{dn_r}{dt} &= n_r n c \int_{W_r}^{\infty} \frac{d\sigma}{dW_s} dW_s = n_r n \frac{2\pi r_o^2 m_e c^3}{W_r} \\ &\approx n_r \frac{e E_{||}}{m_e c \log \Lambda} \end{aligned} \quad (2.2)$$

using Equations (1.1) and (1.2) with the assumption that $E_{||} \gg E_r$. Reference [?] used $v^2/2c^2 = E_r/E_{||}$ as the non-relativistic runaway condition, which accounts for a factor of two difference in formulas.

Although a particular runaway electron can be accelerated to an arbitrarily high energy, the exponentiation in the number of electrons implies the typical runaway has the energy $eE_{||}c\tau_{exp} \approx m_e c^2 \ln \Lambda \sim 10\text{Mev}$, where $1/\tau_{exp} \equiv d \ln n_r / dt$.

The number of exponentiations α_r in the number of runaways from n_{r0} to n_r , which is equivalent to the exponentiation in the runaway current, is

$$\alpha_r \equiv \ln \left(\frac{n_r}{n_{r0}} \right) = \frac{e \int E_{||} dt}{m_e c \log \Lambda}. \quad (2.3)$$

The change in the poloidal magnetic flux outside a magnetic surface that encloses a fixed toroidal magnetic flux [?] is $\Delta\psi_p = 2\pi R_0 \int E_{||} dt$. The number of exponentiations is then

$$\alpha_r = f_R \frac{\Delta\psi_p}{\psi_r}, \text{ where} \quad (2.4)$$

$$\begin{aligned} \psi_r &\equiv 4\pi R_0 \frac{m_e c}{e} \ln \Lambda, \\ &\approx 0.36 R_0 \text{ Volt} \cdot \text{s/m}. \end{aligned} \quad (2.5)$$

The factor $f_R = 2$ in the simple theory outlined here, but the value given by Rosenbluth and Putvinski [?] was $f_R = \sqrt{\pi/3}$ at large aspect ratio with only electron scattering.

The maximum number of exponentiations that could occur in ITER is $\alpha_{max} \approx 30$ when the plasma current is 15 MA—the maximum number of exponentiations is proportional to the current. In existing tokamaks, a thermal quench generally occurs when the current is an order of magnitude smaller than in ITER, so $\alpha_{max} \sim 3$.

Since the runaways are moving along the magnetic field lines at essentially the velocity of light, Equation (2.2) can be rewritten in terms of the current density in runaways

$$\frac{dj_r}{dt} \approx \frac{E_{||} - E_r}{E_r} \frac{j_r}{\tau_r}, \quad (2.6)$$

$$\tau_r \equiv \frac{\psi_r}{2\pi R_0 E_r} \approx 0.85 \text{ s} / n_{20}, \quad (2.7)$$

where n_{20} is the electron number density in $10^{20}/\text{m}^3$. As will be seen in Section ??, the time scale for transferring the current from thermal to runaway electrons is much shorter than τ_r since $E_{||}$ is generally far larger than E_r .

III. CONVERSION OF PLASMA CURRENT TO RUNAWAYS

When runaway electrons carry a negligible fraction of the current, the parallel electric field is $E_{||} = \eta j$, where η is the Spitzer resistivity. The condition for runaways, $E_{||} > E_r$, requires

$$T_e \leq 470 \text{ev} \left(\frac{j_{\text{MeV}/\text{m}^2}}{n_{20}} \right)^{2/3}. \quad (3.1)$$

The current density $j_{\text{MeV}/\text{m}^2}$ has units of MA/m^2 , and the background electron density n_{20} is in units of $10^{20}/\text{m}^3$. In ITER, $j_{\text{MeV}/\text{m}^2}/n_{20}$ can be roughly 3, which means runaways can occur when the electron temperature drops for any reason below about a keV. When ITER has a significantly higher electron temperature, all runaways decay away on a time scale τ_r , Equation (2.7).

The thermal quench that is part of a disruption is expected to cause the electron temperature in ITER to plummet from roughly 20 keV to about 10 eV in a few ms, which could make $E_{||} = \eta j \sim 10^3 E_r$ and initiate a rapid conversion of the plasma current from being carried by thermal to runaway electrons.

To increase the runaway current density by a factor e^{α_r} , the poloidal flux outside of a surface that contains a fixed toroidal flux must change by a factor $\alpha_r \psi_r$. As long as the runaway current is small compared to the total current, the rate at which the poloidal flux changes is determined by the thermal electrons, which means on a time scale $\tau_{L/R} = (a/2.4)^2 \mu_0 / \eta$. The factor of 2.4 comes from the Bessel function that appears in a cylindrical model of the flux diffusion. The resistivity $\eta \approx 2 \times 10^{-5} (10 \text{ev}/T_e)^{3/2}$, and in ITER $\tau_{L/R} \approx 43 \text{ms} (T_e/10 \text{eV})^{3/2}$. More precisely the time for the conversion from a thermal to a runaway current is $\approx (\alpha_r / \alpha_{\text{max}}) \tau_{L/R}$, where α_r is the number of exponentiation that are required and

α_{max} is number of exponentials that are possible; $\alpha_{max} \approx 2I$, where I is the plasma current right after the current quench in MA. The required number of exponentiations in ITER, α_r , is usually assumed to be between five and ten.

The dynamics of the conversion of the plasma current from a thermal into a runaway current with a different radial distribution has been studied in a number of papers [?,?,?].

IV. RUNAWAY SEED CURRENT

The radial distribution of the runaway current density j_r is to a large extent determined by the radial distribution of the initial or seed runaway current, j_s since $j_r = e^{\alpha_r} j_s$.

When the thermal quench is fast, the initial seed of runaways comes from the pre-quench Maxwellian tail [?,?,?], which tends to be highly centrally peaked. To runaway, an electron must have an energy greater than a critical energy H_c , which is determined by two criteria: (1) The electron must have an energy larger than the runaway energy in the electric field, which means $H_c > m_e c^2 E_r / E_{||}$, where E_r is the required electric field for runaways and $E_{||}$ is the actual electric field, Equation (1.2). (2) The electron must have an energy sufficiently large not to have slowed down during the time of the thermal quench τ_q , which is expected to be a few ms in ITER. Since the slowing down time of high energy electrons scales as their velocity cubed, $H_c > (\tau_q / \tau_{th})^{2/3} T_e$ in order not to slow down during the thermal quench, where τ_{th} is the collision time of thermal electrons before the thermal quench and T_e is their temperature. The number density of electrons out of the pre-thermal-quench Maxwellian that satisfy the conditions for runaway is $\approx n e^{-H_c / T_e}$. The collision time of thermal electrons is $\tau_{th} = 6.4 \times 10^{14} T_e^{3/2} / n$ in seconds, where the electron temperature is in keV and the number density is in particles per cubic meter, which in ITER with $n \approx 10^{20} / \text{m}^3$ and $T_e \approx 10$ keV is $\tau_e \approx 0.2 \text{ms}$.

About one electron in ten thousand needs to go to a relativistic energy for runaways to carry the full ITER current. This follows from $j_c \equiv enc = 4.80 \times 10^3 n_{20} \text{ MA/m}^2$, where n_{20} is the electron number density in units of $10^{20} / \text{m}^3$. A typical current density in ITER is a

few MA/m².

The required number of e-folds to transfer the current from thermal to runaway electrons is

$$\alpha_r \approx \frac{H_c}{T_e} - \frac{j}{j_c}. \quad (4.1)$$

V. EFFECTIVE OHM'S LAW

Even when the current is carried by runaways, the deviation in the plasma response from an ideal Ohm's law during a resistive instability is determined by the Ohm's law of the thermal electrons [?]. Since runaway electrons are moving at the velocity of light, the only way to change runaway current is to change the number of runaways, which is over a time scale $\tau_r E_r / (E_{||} - E_r)$, Equation (2.6). There is a long lag time between a change in the electric field and the response of the runaway current. Thermal electrons respond as $\eta \vec{j} = \vec{E} + \vec{v} \times \vec{B}$ on the time scale of their collisions. So deviations from an ideal response, such as the opening of islands, are determined by the thermal electrons.

VI. MICROTURBULENCE

The micro-instability that is thought to be of greatest potential importance to the runaway issue on ITER is an instability of the whistler waves [?,?]. This instability requires a strong anisotropy of the perpendicular to parallel electron momentum in a beam $|p_{\perp}/p_{||}| \ll 1$, and is stabilized quasi-linearly by spreading p_{\perp} , or equivalently the pitch angle ϑ .

Pitch angle spreading can have two effects: (1) The power loss from synchrotron radiation, which is proportional to $\sin^2 \vartheta$, is enhanced [?,?]. (2) The energy gained from the parallel electric field, which is proportional to $\cos \vartheta$, is reduced. Indeed, if pitch-angle scattering were large enough to reverse the direction of electrons along the magnetic field, runaway

electrons would increase their number exponentially in proportion to $\sqrt{t_{scat}t}$ instead of t , where t_{scat} is the pitch-angle scattering time.

Unfortunately, the rate of power loss due to synchrotron radiation ν_s is not large at ITER densities compared to the loss due to the collisional drag of the background electrons ν_c . The ratio for an electron of energy $(\gamma - 1)m_e c^2$ moving with a pitch angle ϑ relative to the magnetic field is

$$\frac{\nu_s}{\nu_c} = \frac{2}{3} \frac{\omega_{c0}^2}{\omega_{p0}^2} \frac{\gamma^2}{\ln \Lambda} \sin^2 \vartheta; \quad (6.1)$$

$$\frac{\omega_{c0}^2}{\omega_{p0}^2} = 0.0974 \frac{B^2}{n_{20}}, \quad (6.2)$$

where ω_{c0}/ω_{p0} is the ratio of the cyclotron to the plasma frequency of the background electrons, B is in Tesla and n_{20} is the electron number density in units of $10^{20}/\text{m}^3$. It is also unfortunate that the quasi-linear stabilization appears to occur [?] when the typical electron pitch angle is small, $\vartheta \sim 0.1$.

A. Uniqueness of the ITER Scale Tokamaks

VII. SCIENTIFIC OBJECTIVES

A. Runaway Electrons Self Consistently Coupled to 3D MHD

B. Long Timescale 3D Evolution of runaway electron current

1. *after thermal collapse, dependence of the final state on initial conditions*

1 page, Brennan, Boozer, Breizman, Fitzpatrick

C. Potential Effect of Microturbulence on Avalanche Mechanism

VIII. PROPOSED RESEARCH AND METHODS

A. Simulating of Runaway Current Generation with XGC1

1 page Ku

B. Simulating of Thermal Collapse with NIMROD

Two key features in the NIMROD code enable the simulation of a thermal collapse. First, the (temperature dependent) anisotropic heat conduction capability is critical for modeling disruption scenarios in which the destruction of flux surfaces leads to large parallel heat transport along stochastic fields from the core to the divertor. Free boundary calculations with very anisotropic thermal conduction have successfully captured the TQ time scale and qualitative features of a beta-limit disruption in DIII-D, and computed the 3D distribution of heat flux to the divertor surfaces [Kruger05]. Second, the option for inclusion of an impurity species, and the associated calculation of impurity radiation allows mitigated disruptions (or rapid-shutdowns) to be simulated, in which the thermal collapse is partially or primarily radiative [Izzo13]. In the case of massive gas injection, with shallow penetration of injected impurities, the loss of core heat was found to be a combination of convection and conduction of heat to the radiating mantle of impurities near the edge, and core radiation following mixing of impurities due to MHD-generated flows.

The rapid-shutdown scenario has special relevance to ITER and the runaway electron (RE) problem. Primarily, this is due to the fact that ITER will tolerate very few unmitigated disruptions, so it is anticipated that rapid-shutdown-type disruptions will be the most common. Also, disruptions for which the primary channel of core electron heat loss is conduction to the divertor will inevitably also have a large associated loss of relativistic electrons to the divertor, thus partially mitigating runaway avalanche growth, at least in

the TQ and early CQ stage; whereas, a radiative thermal collapse can be accompanied by good confinement of core runaway electrons. This fact may partially explain the greater prevalence of large RE populations following mitigated disruptions in present tokamaks, as compared to unmitigated disruptions. Although other factors may also be in play here, and the much larger ITER avalanche term may nullify this distinction and produce large RE currents in any disruption scenario, the experience of smaller tokamaks points to one additional motivation for modeling mitigated disruptions in particular, which is validation. Clearly, for this purpose it will be important to model those scenarios that actually produce observable REs in present machines.

C. Simulating the Thermal Collapse with M3D-C1

1. M3D-C1 Description

M3D-C1 is a code for solving two-fluid magnetohydrodynamics (MHD) equations in toroidal geometry. A semi-implicit timestep is implemented using a physics-based preconditioning method that allows efficient calculations on both stability and transport timescales [?]. Spatial discretization is achieved using high-order triangular finite elements [?] on an unstructured mesh in (R, φ, Z) coordinates, which facilitates free-boundary calculations in diverted magnetic geometry. M3D-C1 has been successfully applied to a range of tokamak phenomena, including the calculation of axisymmetric equilibria with sources [?], linear and nonlinear ELM calculations [?], time-independent response to applied 3D fields [?,?], and repeating sawtooth calculations on current-diffusion timescales [?]. M3D-C1 has been benchmarked through comparisons with other codes [?] and validated against experimental results [?].

A resistive wall model has recently been implemented in M3D-C1. In this model, the resistive wall has a finite width within the computational domain, in contrast to the more common method of using a thin-wall boundary condition at the computational domain

boundary. By including the wall within the domain, the currents driven inductively and electrostatically in the wall can be spatially resolved, and the resulting forces can be easily computed. Additionally, avoiding the boundary-condition method obviates the scalability concerns introduced by the non-local coupling of boundary nodes required by that method.

M3D-C1 is parallelized using MPI. Linear calculations typically require 5–100 CPU-hrs. 3D nonlinear calculations may consume 50k–500k CPU-hrs, depending on the resolution and timescales involved, typically on 2k–8k processes. The semi-implicit time step results in block-tridiagonal matrices, which are solved using a Krylov method (GMRES) with block-Jacobi preconditioning with MUMPS or SuperLU-dist [?]. The linear solvers are implemented using PETSc [?,?,?]. Mesh generation and partitioning is handled using FMDB software.

2. Modeling of Disruptions and Runaway Electrons with M3D-C1

FIG. 1. The toroidal current density in a simulation of a VDE in DIII-D, at (*left*) $t = 2.27$ ms and (*right*) $t = 3.24$ ms. The resistive wall used here approximates the DIII-D first wall, and is shown as the thin (2 cm) region between the blue and purple curves.

Vertical displacement events (VDEs) have been successfully modeled with M3D-C1 in both DIII-D and NSTX. Figure 1 shows the toroidal current density in a simulation of a

DIII-D discharge during an early and late phase of the VDE. In this simulation, M3D-C1 was initialized using a reconstruction of the plasma equilibrium immediately after the thermal quench induced by massive gas injection (MGI) in a DIII-D discharge. In both the experiment and the simulation, the plasma was then observed to become vertically unstable and disrupt. In the simulation, current is observed to flow from the plasma through the resistive wall. These calculations enforce axisymmetry, but nonaxisymmetric calculations are underway that will capture $n = 1$ instabilities expected to occur when the plasma becomes limited by the wall and the safety factor drops.

The effect of runaway electrons (REs) will be included in these calculations by reading the current density due to the REs, \vec{J}_{RE} , as calculated by XGC. From this current density, the magnetic field \vec{B}_{RE} will be calculated by M3D-C1 by solving $\nabla \times \vec{B}_{RE} = \vec{J}_{RE}$, and added to the fields from all other sources (plasma currents, resistive wall, and external coils). The two-fluid MHD equations will then be integrated to advance the densities and pressures of the thermal species, as well as the associated current densities and electromagnetic fields. These fields will be passed to XGC to advance the REs, and this process will be iterated. During the M3D-C1 phase of this iteration, the current density (and associated magnetic fields) of the REs can be held fixed to avoid redundant or inconsistent evolution of the RE current.

Because this iteration is not implicit, there is the possibility of numerical instability. Depending on the type of instability encountered, if any, one of several stabilizing methods could be employed. If a grid-scale instability is found to develop, the RE current density could be artificially smoothed through the application of a hyper-resistive term in M3D-C1. This will improve stability at the expense of fine-scale resolution. If the solution is found to spuriously oscillate in time, a predictor-corrector method can be employed, in which the M3D-C1 / XGC iteration is performed twice (or more) per time step. Since the XGC computation is expected to be significantly more computationally expensive than the M3D-C1 computation, this predictor-corrector iteration could be done with a reduced number of particles in XGC in the first iteration(s).

D. Integrating the MHD and Gyrokinetics through the EFFIS Framework

E. Integrating the MHD and Gyrokinetics through the EFFIS Framework

Our ability to reach the goal of whole fusion device simulations is a large undertaking that involves the development of the appropriate physics models, developing methods to solve this physics models, accounting for the interaction (coupling) of the physics models for whole device simulations, and finally the ability to perform uncertainty quantification with a full linkage to, and interaction with, validation experiments. The level of computational effort required for these simulations can only be addressed on new generations of massively parallel heterogeneous computers where the solution to each physics model scale, and the computational processes that couple these processes also scale, to provide a simulation workflow that will effectively execute and scale. Considering the fact that these multiscale simulation workflows will be executed using a constantly evolving set of models and coupling methods, it is critical that a software infrastructure be provided that will support the effective addition of new models and coupling methods, and provide the means for their execution to take full advantage of the alternative modes of execution on a constantly evolving heterogeneous parallel computing environment. The goal of the End-to-end Framework for Fusion Integrated Simulation (EFFIS) is to provide that software infrastructure.

Our team has previously addressed the challenges of coupling multiple fusion simulations in the Center for Plasma Edge Simulations (CPES), and EPSI SciDAC projects. The overarching objective of our framework is to provide enabling technologies for complex coupled simulations running on leadership class HPC systems. The key focus of our efforts has been on hiding complexity from users by providing an easy-to-use, scalable, and robust computational framework, i.e., the End-to-end Framework for Fusion Integrated Simulations (EFFIS). Our technologies have been applied to enable simulations on over two hundred thousand processors, to help scientists 1) couple multiple codes together using both memory to memory, and file based coupling, 2) monitor their simulations during these runs, and 3)

achieve scalable high-performance I/O. The main technologies that we have deployed in the initial version of EFFIS were: (i) the ADIOS componentized I/O library capable of scaling to hundreds of thousands of cores, (ii) standard scientific workflow templates using Kepler for composing coupled simulations with auxiliary analysis and visualization technologies (iii) the DataSpace framework to flexibly and scalably support in-memory coupling and interactions required by the coupled simulation workflows, (iv) a Dashboard where scientists can quickly access information about their runs in real-time, and finally (v) use of provenance to identify the original files from which visualization products were generated. As the Leadership Class Facilities evolve, and the complexity of the coupling of multiple simulations to multiple analysis and visualization services increased, our team has continued to evolve ADIOS, DataSpaces and the overall EFFIS framework to meet the challenges that are being echoed in Exascale computing. The foundation of our research and development has been in creating an extensible framework that can be used in, and significantly impact, numerous application areas including: Astrophysics, Climate, Combustion, Earthquake Modeling, Flood Prediction, Fusion, Geoscience, Materials Science, Neutron Science, Nuclear Science, Pathology Quantum Turbulence, Relativity, Seismology, Sub Surface Reservoir Modeling, and Weather Modeling.

The further development of EFFIS, EFFIS 2.0, as part of this project will build directly upon ongoing ASCR supported efforts on architecture aware tools and technologies needed to support whole fusion device simulations with specific linkages made to the SDAV, FASTMath and SUPER SciDAC Institutes. EFFIS 2.0 will provide the overall software framework for coupled simulation workflows, integrating methods that can efficiently couple the physics models accounting for the form of discretization used, and address issues related performance optimization and scaling.

1. Core technologies for EFFIS 2.0

The current challenges that we face in our research and development are in creating a sustainable software framework that can play a major part of exascale computing. As projects such as EPSI evolved, so has EFFIS, and our research and development has been pushing us towards EFFIS 2.0 which will be the focus of this project. Our EFFIS 2.0 effort, in general, explores how to make it easy to compose coupled simulations with workflow technologies while the framework is capable of moving data between coupled codes efficiently, and how to debug what is happening during coupling. EFFIS 2.0 will include extensions to DataSpaces to allow coupling of plug-ins (codes, visualization services, etc.) on the same node using memory references, or memory copies, on different nodes on the same machine using either RDMA, MPI, or sockets, or on different machines when security of the system can allow this. We will also work with the applied math groups within the team to tightly integrate the new math services into the EFFIS 2.0 framework, which will ensure that data is being passed properly from one plugin to another plugin. By integrating the performance information, and a well-defined schema, we will allow performance engineers the ability to quickly determine when performance problems exist in the codes.

The new additions we propose to enhance EFFIS 2.0 in this proposal are: 1) Tighter integration of the provenance capturing routines to the ADIOS data streams which can be used for in-situ or post-partum forensics of coupled codes, 2) New data movement abstractions which can use MPI communication for greater flexibility across the wide range of DOE platforms and DataSpaces for low latency, high bandwidth code coupling, 3) New analysis and visualization services which can be flexibly plugged in into a coupled simulation and 4) Integration of the ITER ITM schema into EFFIS 2.0 to allow for greater collaboration of data from this project to the greater fusion community.

Tighter Integration of Provenance in EFFIS

One of the keys to reproducibility as well as to the usability of these technologies is the provenance and metadata capturing system, which allows for gathering provenance infor-

mation from across all the applications and analysis and visualization plugins. Provenance enables users to identify for any image plot the source code of the simulation, the build dependencies of the simulation where it was executed and the input arguments to the simulation run that produced the dataset from which the image was created from. EFFIS 1.0 had a provenance capturing system implemented at the Kepler workflow level, that recorded the input and output command-line arguments of stand-alone executables in a file-based processing of data. This was sufficient for ensuring reproducibility in the CPES project. It also used an external relational database to store all provenance information, separately from the datasets.

EFFIS 2.0 is targeting a much more complex computing environment where data is moved as streams among coupled as well as standalone components and where operations on the data cannot be captured completely by an external entity like a workflow execution engine. Provenance metadata needs to be generated on the spot by the EFFIS toolkit where the action happens, and needs to be carried along with the data itself. Since the ADIOS programming API is the interface between any of these components and is the interface of all data flowing through the processing pipelines or between coupled applications, it is the natural place to put a provenance capturing interface as well. ADIOS can insert the collected provenance information into the data stream as metadata, which is already extensive to support self-describing datasets augmented with a visualization schema and possibly going through data transformation (like compression) steps inside ADIOS.

In this proposal we propose to create a provenance capturing interface in ADIOS, the extension of the ADIOS metadata structure and the mechanism inside ADIOS to insert the captured information into the metadata. Analysis and visualization services provided as part of the EFFIS toolkit will be extended to provide the provenance information.

Data movement abstractions for coupling

ADIOS provides the I/O API for applications to declare their input and output activities. It provides various I/O strategies for data operations that allow applications to adapt across platforms/architectures and choose the strategy that best fits their requirements.

In particular, ADIOS offers a range of staging I/O methods under the same API, so that applications can perform memory-to-memory data exchange on a single computing system without touching the relatively slow disk-based file-system. Newly developed methods can transfer data over the wide-area network, or through cloud services such as those provided by Amazon. In this effort we propose to develop two new capabilities for data movement to support coupling.

First, we propose to implement an MPI-based staging method that allows for moving data between two subsets of application processes running under one MPI world communicator. The key advantages of coupling codes using MPI is that MPI has been well developed and, more importantly, optimized on all commonly used Interconnects and a newly install HPC machine will have MPI support by default. This implementation will also provide a fallback strategy for new application.

Second, we will build on ADIOS/DataSpaces and will extend it to support the in-situ, in-memory coupling required by the targeted coupled fusion simulation workflows. Specifically, we will extend the DataSpaces framework and define semantic mechanisms to allow coupled applications to join and leave the shared space without affecting other applications. The resulting DataSpaces as a service will allow applications join and leave shared space while DataSpaces staging servers keep running as a persistent service. In this effort we will also explore runtime services to co-locate coupled codes on the same compute nodes and to facilitate in-memory data sharing using on-node shared memory segments.

Visualization of coupling data

EFFIS provides a highly efficient framework to exchange datasets in a coupled application where data lives only temporarily and is being transformed in transit from one code to the other. It is of utmost importance to be able to examine these datasets to understand if the coupling is done right. We need to be able to compare a dataset in one code before the coupling and the same dataset in the other code after transformation and transfer. Comparative visualization of the two helps to locate the source of a bug.

In EFFIS, we envision a large range of analysis and visualization operations, including

renderings of particular subsets of the data (e.g., slice, clip), feature detection and extraction (e.g., islands, X,Y), comparative visualization, and whole data visualizations (e.g., volume rendering, flow field rendering, etc). These services will be deployed as plug-ins to the EFFIS 2.0 framework that can be assembled together as required by the users to examine the coupled code behavior instead of a single built-in visualization built for one purpose.

To enhance the interface between application scientists and the visualization experts we will heavily leverage, and expand upon the ADIOS Visualization Schema. The Viz Schema allows applications scientists to provide a semantic description of the particular content of data streams. This allows visualization services the ability to properly understand the structure, relationships and meanings of the data. To better support code coupling, comparative visualization and the issues of meshes, we will also adding information into the viz schema that will assist in coordinate and mesh transforms, such as Jacobians.

To enable these analysis and visualization plug-ins to work well in an exascale environment, we will be leveraging and integrating several key technologies being developed by the SDAV Institute into the EFFIS 2.0 framework, including VisIt for off-line visual exploration of datasets and VTK-m to develop a platform that is suitable for the emerging HPC resources. The light-weight, advanced data model, and execution environments in VTK-m make it ideal for plugin-ins within the EFFIS 2.0 framework that will allows us to produce visualizations on the fly where the coupling happens.

Integration of the ITER ITM data model into EFFIS 2.0

EFFIS aims to provide a generic toolkit for extreme scale computing composed of various applications, services and plugins and for running those together flexibly and efficiently on large scale computing resources by any domain scientist. However, some aspects of handling of data are domain specific. In this project we want to enable greater collaboration of all data generated from this project to the greater fusion community. This is very important for the sustainability of the project and of EFFIS, so that the entire fusion community can benefit from the results of this project by being able to take the generated data and use in their respective applications. A common data model for the scientific data objects of a

domain is therefore of great importance.

The ITER Physics Data Model (IDM) is under development by European fusion groups and the EPSI project in the U.S. decided to incorporate this model. We believe it is beneficial to have other fusion applications be able to use the ITER data structures and therefore we will rely on the definition of data structures of IDM.

We will develop an interface to exchange physics objects between codes (e.g. core profile, equilibrium) in a precisely defined manner. The interface will be implemented on top of ADIOS, so that the exchange in coupled runs as well as their permanent storage is efficient. These approach will enable the project to be even more flexible, as new codes can be added to the project later and easier if all codes use the same data model. This makes the transformation of data from one code's output to be input to another straightforward.

2. Coupling of runaway current generation and thermal collapse analyses in EFFIS 2.0

The FASTMath Institute's [?] tools and expertise related to scalable infrastructures for mesh-based simulation methods, including the coupling of mesh-to-mesh and particle-to-mesh methods, will represent the starting point for developing the coupling processes needed for the multiscale simulation of tokamak disruptions with kinetic runaway electrons. Building on its core parallel mesh infrastructure tools, FASTMath is already supporting mesh related requirements of the XGC1 and M3DC1 codes as follows:

1. Employing the Parallel Unstructured Mesh Infrastructure (PUMI) [?] to provide general unstructured meshes (2-D and 2.5D (extruded 2-D)) meshes for M3DC1 [5,4].
2. Providing mesh adaptation methods for M3DC1 [5].
3. Supporting the internal operations in M3DC1 to go from PUMI mesh data to the identification of the PETSc global systems and supporting assembly of element matrices.
4. Providing XGC1 new mesh generation procedures that are more automated, faster and more flexible, which maintaining mesh configurations constraints.

5. Working on providing parallel mesh representation building off of PUMI to be couple the XGC1 field and particle level analysis procedures.

The above developments are taking advantage of FASTMath related mesh developments that include:

1. A parallel mesh infrastructure (PUMI) [3] that fully support evolving meshes [?] with recent extensions to run in a hybrid MPI/threading mode [9]. PUMI has been used to support adaptively defined meshes of 92 billion elements analyzed on 3/4 compute cores [12].
2. Parallel mesh adaptation procedures capable of performing parallel general mesh modifications on highly anisotropic mixed meshes maintaining semistructured regions [11,13].
3. Attached parallel fields [1] to support the relating of fields to meshes.
4. Methods for fast dynamic load balance of meshes to ensure the scalability of component simulation workflow operations [?,2,18].
5. Methods to perform the in-memory coupling of mesh-based simulation components [16].

The technical requirements of coupling the runaway current generation analysis to be performed by XGC1 and the various regions of potential thermal collapse using NIMROD and M3DC1 begins by identifying the fields that need to be coupled. Although the bulk of the XGC1 calculations involve the particle level tracking of ions and electrons, the particle information is used in conjunction with a field solve on an unstructured mesh to define the 3D current density field on that mesh. This 3D current density field is to be used as a forcing function in the mesh-based magnetohydrodynamics (MHD) flow analyses performed by NIMROD and M3DC1 which calculate magnetic and electric fields to be used as forcing functions in the mesh-based field solve in XGC1. Therefore, the performance of the coupled

simulation requires the mesh-to-mesh coupling of these fields not only at the level of the advancement in time of the simulation, but within the individual non-linear steps.

Core considerations in the development of a mesh-to-mesh coupling method are (i) satisfaction of consistency requirements, (ii) ensuring convergence, (iii) accuracy and (iii) computational efficiency. Consistency relate to requirements of the physics models being solved. Consistency can relate to basic conservations requirements such as conserving total mass or charge where there is an absolute requirement to satisfy the property on a global bases and strong desire (in terms of maintaining rates of non-linear convergence) in satisfying them on as local a basis as possible. An example of maintaining properties locally is the use of local solution transfer during mesh adaptation [17]. A divergence free magnetic field must be maintained as it is transferred from an MHD mesh to a XGC1 mesh while a divergence free current density must be maintained as it is transferred from XGC1 to the MHD codes. Irrespective of the order of accuracy (convergence rate) of the mesh-based discretization, a minimal requirement of the solution transfer procedure is to ensure at least first order convergence of the solution transfer so the the entire process converges. Of course in the cases where higher order methods are applied, as in both M3DC1 and NIMROD, employing a solution transfer method of lower order is wasteful of computations done on the higher order mesh. Thus from an accuracy and computationally efficiency perspective the solution transfer method should match the accuracy that is needed to maintain the rate of convergence of the mesh-based discretization method. Of course, as is the case here-in, this becomes a more complex issue when the two meshes being coupled have different rates of convergence (more on this below). The other key aspect of the computational efficiency is associated with determining, and maintaining, the mesh-to-mesh relationships, particularly in parallel.

All three codes employ a 2D axisymmetric mesh discretized by piecewise basis function in the poloidal plane of the reactor, with specific constructs in the third direction. Thus, the interactions of the 2D meshes are the central consideration in the mesh-to-mesh transfer process. However, the 2D meshes used by each of the codes are different in multiple ways. Both XGC1 and M3DC1 use unstructured meshes of triangles. However, the XCG1 mesh is

based on a piecewise linear C^0 elements that are carefully aligned to match the magnetic flux field. The M3DC1 mesh is based on a C^1 high order discretization that is complete of fourth order. NIMROD employs a high order block structured grids of primarily quadrilateral elements. In all cases one of the meshes is a fully-unstructured meshes and any mesh-to-mesh transfer process with an unstructured mesh will need to define a effective searching structure to support the mesh-to-mesh point location problem. The fact that different orders of spatial discretization are employed in each of the three meshes means that the level of spatial mesh resolution for a given level of solution accuracy for just the basic transfer operation will need to be substantially different which will complicate the development of an effective solution transfer process in terms of both determining the spatial resolution needed for nearly equal accuracy of the fields shared between meshes and complexity of the mesh searching algorithms. If nearly equal accuracy of the fields shared between meshes is not attained, the more accurate mesh calculation will have wasted computing resources obtaining that higher level of accuracy. Search structures that are well suited for mapping between meshes of nearly equal spatial resolution may be less than ideal for mapping between meshes of substantially different spatial resolution. The fact that the different meshes use different spatial discretization methods and orders will introduce complications in the applications of the projection and interpolation methods to be applied with respect to maintaining consistency and attaining the best accuracy.

Given the requirements and complexities of the mesh-to-mesh solution transfer process in coupling XGC1, NIMROD and M3DC1, developments required will include: (i) Parallel methods to support point location for distributed meshes. (ii) Defining projection and interpolation methods appropriate for each of the discretization methods that satisfy the consistency requirements and address the accuracy of the transfer. (iii) Controlling the mesh resolution of the coupled meshes using adaptive mesh control methods that now included maintaining equivalent accuracy between the meshes being coupled. (iv) Integrating of the procedures developed into EFFIS. (v) Developing of strategies to execute the mesh-to-mesh solution transfers in a manner in which the complete coupled simulation scales and effectively

used the resources of the parallel computing system.

Supporting the point location - PUMI for the mesh, need a search structure to relate parallel distributed meshes - need to consider that fact the XGC1 mesh is likely to be much finer than the NIMROD and M3DC1 meshes,

Projection and interpolation - Driven first by the consistency issues, then accuracy

Adaptive control of mesh resolution - The standard one is mesh adaptation to control the discretization errors for the fields to be transferred - In the case of XGC1 this would need to be done while still maintaining the mesh constraints needed for effective coupling with the particle methods. The second adaptive control is trying to have the same basic level of accuracy between the meshes for the different codes - likely not too hard to get something reasonable, but it is important.

I expect that there will need to be extensions to EFFIS to deal with issues associated with having multiple related meshes on the same domain. I expect this since when everything is related to a single mesh EFFIS can be pretty blind to the meshes and relations of fields between them. When there are multiple meshes there will be a minimum of some high level information to say they, and their fields, are related. However, there may be more to it in which case getting a bit tighter linkage with the mesh (PUMI) and fields (APF) may be important.

The key issues associated with the mesh-to-mesh execution strategy relate to how the computations are distributed over the parallel computer and to performing dynamic load balancing to gain scalability. The working assumption is that EFFIS will support the ability to distribute the calculations on the different meshes over the nodes and cores in different manners while supporting the movement of information. Dynamic load balancing as things are adapted and as operations change can be supported by tools like Zoltan and ParMA.

3. Performance Engineering for EFFIS 2.0

Achieving good performance in coupled simulations is a daunting and a labor-intensive task. In this project, the University of Oregon team will deploy the TAU Performance System [15] for performance engineering, verification, and validation tasks. TAU provides a comprehensive profiling and tracing toolkit for performance evaluation and tuning of HPC codes. It supports the source instrumentation of memory, I/O, and communication libraries.

To fully observe the performance artifacts affecting the coupled MHD simulation running on leadership class DOE heterogeneous HPC platforms, it is necessary to measure the performance of individual components and the EFFIS framework connecting the XGC1, NIMROD, and M3D-C1 codes. Parts of these applications will execute on accelerators that share memory with the compute nodes. Instrumenting applications that use NVIDIA and AMD GPUs and Intel Xeon Phi co-processor (aka MIC) accelerator nodes requires a common performance model in a tool that supports this diversity of accelerator devices in its instrumentation and measurement layers [10]. As the coupled MHD codes are instrumented and executed on these hybrid nodes, data exchange between a host CPU and an accelerator connected by a slower PCI-X bus, and between compute nodes themselves will require careful orchestration and tracking of the asynchronous execution of kernels. When the instrumented kinetic and MHD components are executed concurrently in different spatial grids and computational nodes, common performance instrumentation events from multiple layers will flow into a common performance data repository in TAU. This data will be exported into profile files for subsequent analysis as well as stored as performance provenance data records in the output files collected at the end of the execution.

The ADIOS layer will be instrumental in exposing the application data and the performance data for each execution. By providing the TAU performance provenance information as a component of the ADIOS output, it will help us classify and explain the variability in I/O performance due to system noise and use of shared I/O resources. TAU will interface with ADIOS to extend the current scope of the output data to retain this critical

information. Profiling the I/O resources of the simulation [14] will also help shed light on data exchange that will be optimized as we transition from a file system based approach to an in-memory approach that uses MPI and other technologies such as Kepler and Swift in the EFFIS 2.0 framework. The data transfer rate between codes residing on different spatial grids and computational nodes will be tracked using TAU to help optimize the parallel mesh-to-mesh coupling and interpolation. The performance data will be stored in a TAUdb database [7] and used for scaling studies using TAU's PerfExplorer cross-experiment analysis tool [6,8] and the ParaProf 3D profile browser.

Historical performance data will help us evaluate the efficacy of changes in the EFFIS 2.0 framework during the developmental stage. Specifically, we will evaluate and store performance data collected from the coupled routines, the framework used for coupling, and the I/O layer. The data will comprise of profiles that show the exclusive time spent at the routine, loop, and statement granularity and will track the volume of data. Runtime analysis of data will help identify peaks in the bandwidth of mesh exchanges and the I/O data volume by triggering context events that couple the calling context in the form of a program callstack with the value of the data that exceeds previously seen thresholds. These peaks will reveal fluctuations in performance and the code regions that are responsible for it. When key algorithmic changes are made to the coupling of these codes such as using the same mesh geometry between two components of the simulation, TAU will help quantify the effect of the change by comparing the baseline performance data with the updated components.

F. Runaway Current Flow through a 3D Evolving Magnetic Topology

1page

G. Validation of Physics Elements

Runaway-related physics has been studied experimentally on a number of tokamaks, both during disruptions and during quiescent flattops. The latter cases have the advantage

of being well-diagnosed and reproducible, while the former are more closely related to the overarching theme of this proposal. These studies provide numerous opportunities for objective comparison with the simulations to be done, potentially providing some measure of validation of the physics coming out. Specific examples include measurements of the temporal behavior and spatial characterization of runaway plateaux (growth/decay, amplitude, position, shape, motion) during disruption current quenches, and runaway onset threshold conditions (E-field, density) during quiescent near-steady-state plasmas. In addition, some RE mitigation studies have been, or are in the process of being done, and the results could provide further possibilities for validation. These include impurity injection (gas and pellets) into both post-thermal quench disruptions and quiescent flattops, with the key measurement being the effect on the RE growth/decay rate.

Hollman, Gerhardt, Granetz

1. Hollman

Validation efforts will focus on the runaway electron model and its coupling, as the MHD codes have been separately validated extensively. This is good, because comparison of the MHD structure during the TQ to experimental measurements is a challenge for validation because the edge magnetic fluctuation coils do a very poor job of describing core MHD structure during the TQ. Never-the-less, comparison of the experimental magnetic fluctuation signals to synthetic diagnostics in the MHD simulations will form an important part of our validation efforts.

A few target areas where experiments are expected to provide a validation basis to test MHD + kinetic simulations are (REWORD THESE):

- 1) prompt RE loss number/energy - When we make a disruption with RE formation (which we do in D3D with Ar pellet injection) some REs are lost in the end of the TQ to the wall (the rest avalanche and form the RE plateau). The number/energy of these prompt loss REs provide a test for TQ MHD simulations because they could give information on

the fraction of field lines destroyed and also on the electric fields seen by the REs during the TQ.

2) RE plateau instabilities - we do see occasional MHD instabilities in the RE plateau (characterized by a increasing Bdot signal on the wall and then culminating in a sudden flash of HXR's when part of the REs are lost to the wall). Nobody knows what these are, but this might be interesting to simulate as it would require a coupling of MHD to kinetic simulations as you are planning on doing and we do have some data on this already (wall Bdot signals, HXR signals) so it is something that could easily put into the proposal.

3) final RE loss instability - This is not presently in your outline, but might be worth considering. When a RE plateau hits the wall, there is high toroidal peaking characteristic of a kink and often multiple repetitive HXR flashes indicating a repetitive instability. Nobody has done any simulations of the MHD mode activity of a RE beam hitting the wall, so this would be entirely new turf, but very interesting for ITER and something we could actually give you some data on.

Need to mention JET experiments.

H. Towards Predictive Simulations of Mitigation in ITER

The thermal collapse in ITER is expected to be fast, on the order of 10-20ms. (To avoid significant forces on the wall?) the current quench will be held off to 100-150ms. This will leave significant time for runaway generation. A significant effort is therefore planned for intervention and mitigation of the runaway electron generation. This plan focusses on massive gas and pellet injection during the thermal quench, to generate higher density, fluctuations and in any way possible scattering of the electron beam. Neon or Beryllium gas are thought to be the best choices, with Hydrogen or Deuterium also considered, where Beryllium may have an advantage as it is expected to leave a higher T_e after the thermal quench. This would reduce the loop voltage and thus reduce the runaway generation. Pellet injection is being considered, using the same substances; N, B, H, and D, as the penetration

may be significantly deeper, and the accuracy of deposition advantageous on the relatively thin runaway beam.

It is not expected that the Rosenbluth density limit will be reached under these scenarios, which instead rely on scattering from fluctuations (and radiation). However, experimental evidence suggests that the actual density limit may be lower than the Rosenbluth limit. It is a focussed goal of this proposed research to quantitatively asses this critical question.

IX. MANAGEMENT PLAN AND TIMELINE OF ACTIVITIES

2 pages All of us, but this is a short section **Year 1:** Develop

Year 2: Determine

Year 3: Continue

X. SUMMARY

1 page TBD

The proposed research is focused on

References

- [1] Attached parallel fields webpage. <http://redmine.scorec.rpi.edu/projects/mathfield/>.
- [2] ParMA web page. http://redmine.scorec.rpi.edu/projects/parma/wiki/ParMA_Concepts.
- [3] PUMI web page. <http://redmine.scorec.rpi.edu/projects/pumi>.
- [4] M. Ferraro, L.L. Lao, A.D. Turnbull, S.C. Jardin, S.A. Sabbagh, M.S. Shephard, , and F. Zhang. Resistive wall model in m3d-c1. In *2014 International Sherwood Fusion Theory Conference*, 2014.
- [5] N. Ferraro, S. Jardin, M.S. Shephard, A. Bauer, J. Breslau, J. Chen, F. Delalondre, X. Luo, and F. Zhang. Fluid modeling of fusion plasmas with m3d-c1. In *SciDAC 2011 Conference Proceedings*, pages 69–73, 2010.
- [6] K.A. Huck and A.D. Malony. Perfexplorer: A performance data mining framework for large-scale parallel computing. In *Supercomputing, 2005. Proceedings of the ACM/IEEE SC 2005 Conference*, pages 41–41, Nov 2005.
- [7] Kevin A Huck, Allen D Malony, Robert Bell, and Alan Morris. Design and implementation of a parallel performance data management framework. In *Parallel Processing, 2005. ICPP 2005. International Conference on*, pages 473–482. IEEE, 2005.
- [8] Kevin A Huck, Allen D Malony, Sameer Shende, and Alan Morris. Knowledge support and automation for performance analysis with perfexplorer 2.0. *Scientific Programming*, 16(2):123–134, 2008.
- [9] Dan Ibanez, Ian Dunn, and Mark S Shephard. Hybrid mpi-thread parallelization of adaptive mesh operations. *Parallel Computing*, 2014. submitted.
- [10] Allen Malony, Sameer Shende, Wyatt Spear, CheeWai Lee, and Scott Biersdorff. Advances

- in the tau performance system. In Holger Brunst, Matthias S. Mller, Wolfgang E. Nagel, and Michael M. Resch, editors, *Tools for High Performance Computing 2011*, pages 119–130. Springer Berlin Heidelberg, 2012.
- [11] Aleksandr Ovcharenko, Kedar C. Chitale, Onkar Sahni, Kenneth E. Jansen, and Mark S. Shephard. Parallel adaptive boundary layer meshing for cfd analysis. In Xiangmin Jiao and Jean-Christophe Weill, editors, *Proceedings of the 21st International Meshing Roundtable*, pages 437–455. Springer Berlin Heidelberg, 2013.
- [12] Michel Rasquin, Cameron W. Smith, Kedar Chitale, E. Seegyoung Seol, Ben Matthews, J. Martin, Onkar Sahni, Raymond Loy, Mark S. Shephard, and Kenneth E. Jansen. Scalable fully implicit finite element flow solver with application to high-fidelity flow control simulations on a realistic wind design. *Computing in Science and Engineering*, 2014. accepted.
- [13] O. Sahni, A. Ovcharenko, K.C. Chitale, K. E. Jansen, and M. S. Shephard. Parallel anisotropic mesh adaptation with boundary layers for automated cfd simulations. *Engineering with Computers*, 2013. submitted.
- [14] Sameer Shende, Allen D. Malony, Wyatt Spear, and Karen Schuchardt. Characterizing i/o performance using the tau performance system. In *PARCO*, pages 647–655, 2011.
- [15] Sameer S. Shende and Allen D. Malony. The tau parallel performance system. *Int. J. High Perform. Comput. Appl.*, 20(2):287–311, May 2006.
- [16] C.W. Smith, K. Chitale, D. Ibanez, B. Orecchio, E.S. Seol, O. Sahni, K.E. Jansen, and M.S. Shephard. Building effective parallel unstructured adaptive simulations by in-memory integration of existing software components. *SIAM Journal on Scientific Computing*, 2014. in preparation.
- [17] J. Wan, S. Kocak, and M.S. Shephard. Automated adaptive 3-d forming simulation process. *Engineering with Computers*, 21(1):47–75, 2005.
- [18] M. Zhou, O. Sahni, T. Xie, M.S. Shephard, and K.E. Jansen. Unstructured mesh partition im-

provement for implicit finite element at extreme scale. *Journal of Supercomputing*, 38(3):140–156, 2012. DOI:10.1007/s11227-010-0521-0.